

Praktikum aus Softwareentwicklung 2

WS 2006/2007, LVA 365.009

Abgabetermin: Mo. 15. 12. 2006, 17:00

Erreichbare Punkte: 24

Übung 5 – Java Servlets

Aufgabe 1: Servlet Infos

12 Punkte

Implementieren und Erweitern sie das SnoopServlet, welches diverse Informationen über den Request und den Container anzeigen soll. Verwenden sie die auf der Praktikumswebsite zur Verfügung gestellte Webapplikation und erweitern sie das SnoopServlet um folgende Funktionalitäten:

- Request Information: Req. URL, Http-Header, Req. Parameter, Client IP und Host, Datum und Zeit des Request, Request Methode (POST oder GET), QueryString bei Methode GET, Request Schema und Protokoll
- Server und Servlet Informationen: Servername, Serverport, Servlet Pfad, Echter Pfad mit Context, Servlet Context Name, Container Name, Servlet Version

Ihr Servlet soll eine Startseite mit 2 Links beinhalten, welche die beiden oben beschriebenen Information abrufen. Zur Steuerung der Anzeige der beiden Informationsseiten verwenden Sie keine RequestParameter sondern Path-Informationen. Zum Beispiel könnte ihr SnoopServlet mit folgenden 3 Requests aufgerufen werden:

- <http://localhost:8080/myWebapp/snoop> --> liefert die Startseite, mit den Links auf die beiden Funktionalitäten
- <http://localhost:8080/myWebapp/snoop/do1> --> liefert die Request Informationsseite
- <http://localhost:8080/myWebapp/snoop/do2> --> liefert die Server und Container Informationsseite

Achten Sie darauf, dass auch die Informationsseiten wieder Rücklinks auf ihr Startseite beinhalten. Die Path-Informationen können Sie mit der Methode: `request.getPathInfo()` abrufen.

Verwenden sie die auf der Praktikumsseite zur Verfügung gestellte Webapplikation „myWebapp“ und erweitern sie diese. Bitte geben sie außer dem Sourcecode auch die gesamte Webapplikation ab. Falls sie Änderungen an einem der Tomcat Konfigurationsfiles vornehmen, dokumentieren sie diese.

Auch das Servlet der 2. Aufgabe soll in der Webapplikation „myWebapp“ laufen. Ändern sie dementsprechend die web.xml Datei. Es sollte eine allgemeine Startpage für die Webapplikation geben „index.html“ auf der alle nötigen Links vermerkt sind.

Anbei noch ein Beispiel für die Informationen die auf beiden Seiten angezeigt werden sollen. Trennen Sie die einzelnen Informationen jedoch nach Request-Informationen und Server-(Container)-Informationen damit die Informationen übersichtlich auf 2 Seiten dargestellt werden können.

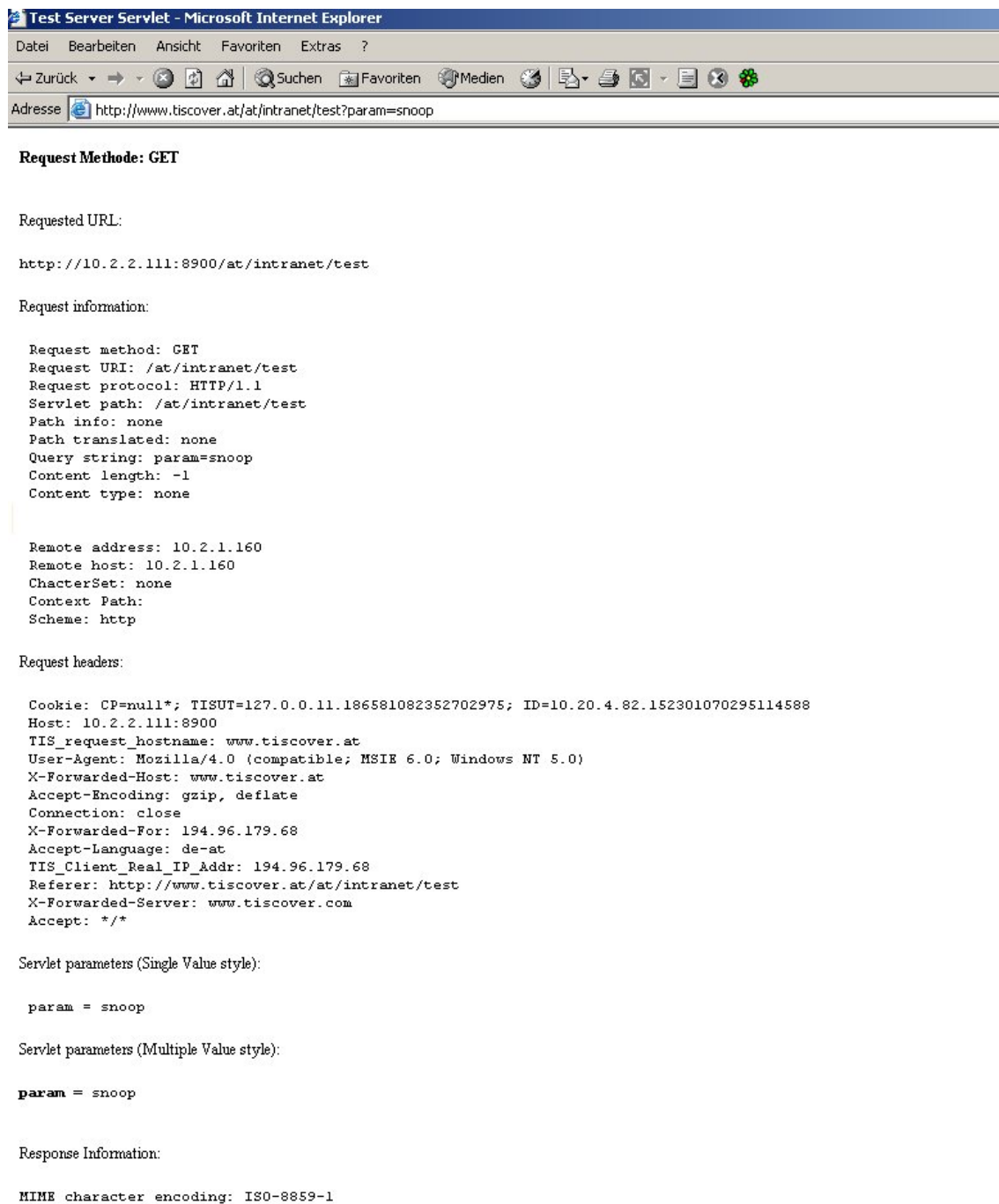


Abbildung 1: Request Informationen

Aufgabe 2: Image-Banner

12 Punkte

Entwickeln Sie ein Servlet, das bei jedem Aufruf ein neues Bild aus einem Verzeichnis des Web-Servers zurückliefert. Implementieren Sie auf diese Art und Weise einen einfachen Werbebanner-Mechanismus. Das Verzeichnis der Bilder wird über einen

RequestParamer angegeben. Verwenden sie das Sessiontracking API um den „Zustand“ des Clients zu speichern (Jeder Client soll seine eigene Reihenfolge haben).

Bsp.: Am Web-Server liegen im Verzeichnis c:\temp\images folgende Bilder:

- img01.gif
- img02.jpg
- img03.gif

Beispiel HTML-Seite:

```
<html><head><title>Test</title></head>
<body></body>
</html>
```

Beim ersten Aufruf dieser HTML-Seite soll das Bild "img01.gif" angezeigt werden, danach "img02.gif", dann "img03.gif" und dann wieder "img01.gif" usw.

Für Debugging Zwecke benötigt das Servlet noch einen Init-Parameter der im web.xml gesetzt wird. Dieser Parameter soll nur für dieses eine Servlet gültig sein. Bsp.:

```
<init-param>
  <param-name>debug</param-name>
  <param-value>true</param-value>
</init-param>
```

Dieser Parameter bewirkt, dass 2 Header Parameter gesetzt werden:

FileName – Der Dateinamen der gerade angezeigten Datei

ChangeDate – Das letzte Änderungsdatum der Datei

Sie können die Richtigkeit dieser Response-Header mit 2 frei verfügbaren Tools ansehen:

InternetExplorer → ieHTTPHeaders
(<http://www.blunck.info/iehttpheaders.html>), Firefox → webdeveloper
(<http://www.chrispederick.com/work/firefox/webdeveloper/>)

Tragen Sie das Servlet in ihre Webapplikation so ein, dass es mit dem Url-Pattern /imgbanner/* aufgerufen werden kann. Bitte testen Sie ihr Servlet auf jeden Fall mit folgenden aufrufen und versichern Sie sich, dass bei falschen oder sinnlosen Aufrufen eine entsprechende Fehlermeldung angezeigt wird.

<http://localhost:8080/myWebapp/imgbanner?basedir=c:\temp\images>
<http://localhost:8080/myWebapp/imgbanner?basedir=>
<http://localhost:8080/myWebapp/imgbanner?base=c:\temp\images>
<http://localhost:8080/myWebapp/imgbanner?basedir=asdasdh>
<http://localhost:8080/myWebapp/imgbanner>
http://localhost:8080/myWebapp/imgbanner?basedir=c:\tem?*asd/asd
<http://localhost:8080/myWebapp/imgbanner?basedir=c:\temp\hallo.gif>

Hinweis: Ändern Sie den ContentType der Servlet-Ausgabe mit res.setContentType("image/gif") oder res.setContentType("image/jpg") und senden Sie einen entsprechenden Byte-Datenstrom (das Bild selbst) als Servlet-Ausgabe zurück. Setzen Sie auch die richtige „ContentLength“, dazu übergeben Sie der entsprechenden Methode des Servlet-API die Größe des Files.